CLAIMS

1. (original) A computer-implemented method for switching between multiple implementations of a routine in a library of routines that are linked with an application program that is hosted by a computer system, comprising:

compiling a plurality of implementations of a routine into respective object code modules, the routine having an associated name and each implementation adapted to a selected hardware configuration;

associating the object code modules with the name of the routine and respective sets of hardware characteristics; and

resolving when the application program is loaded into memory of the computer system, a reference to the routine using the sets of hardware characteristics and a hardware configuration of the system.

2. (original) The method of claim 1, further comprising establishing a symbol table having a plurality of entries, each entry including a name of a routine and a reference to an object code module in the library.

3. (original) The method of claim 2, further comprising, for the routine having a plurality of implementations, adding a plurality of entries to the symbol table and associating respective sets of hardware characteristics with the plurality of entries.

4. (original) The method of claim 3, wherein the hardware characteristics include at least one of clock speed of the processor, processor model, cache configuration of the system, hardware operation latency times, instruction set characteristics, bypass characteristics, branch prediction behavior, pre-fetching capability, information describing stall conditions, branch penalties, size and associativity of processor data structures, queue sizes for out-of-order or decoupled processors, and the number of processors in a multi-processor system.

5. (original) The method of claim 4, wherein the resolving step further comprises obtaining the hardware configuration of the system from at least one of a system configuration data file, one or more system identification registers, and system firmware.

6. (original) The method of claim 3, wherein the resolving step further comprises obtaining the hardware configuration of the system from at least one of a system configuration data file, one or more system identification registers, and system firmware.

7. (original) The method of claim 1, wherein the hardware characteristics include at least one of clock speed of the processor, processor model, cache configuration of the system, hardware operation latency times, and instruction set characteristics.

8. (original) The method of claim 1, wherein the resolving step further comprises obtaining the hardware configuration of the system from at least one of a system configuration data file, one or more system identification registers, and system firmware.

9. (original) A computer-implemented method for switching between multiple implementations of a routine in a library of routines that are linked with an application program hosted by a computer system, comprising:

establishing a set of hardware configuration characteristics that describe the computer system;

establishing a symbol table, the symbol table having one or more entries that include a name of a routine, a set of hardware characteristics, and an address referencing a routine in the library;

obtaining a name of a routine having multiple implementations when the library is loaded with the application program into memory of the computer system;

matching the name of the routine and the set of hardware configuration characteristics that describe the computer system to an entry in the symbol table; and generating an address in executable code for references to the routine having multiple implementations when the library is loaded with the application program, the address referencing an implementation in the library as identified in the matching step by the entry in the symbol table.

10. (original) The method of claim 9, wherein the hardware configuration characteristics include at least one of clock speed of the processor, processor model, cache configuration of the system, hardware operation latency times, and instruction set characteristics.

11. (original) The method of claim 10, wherein the resolving step further comprises obtaining the hardware configuration of the system from at least one of a system configuration data file, one or more system identification registers, and system firmware.

12. (original) The method of claim 9, wherein the resolving step further comprises obtaining the hardware configuration of the system from at least one of a system configuration data file, one or more system identification registers, and system firmware.

13. (original) An apparatus for switching between multiple implementations of a routine in a library of routines that are linked with an application program that is hosted by a computer system, comprising:

means for compiling a plurality of implementations of a routine into respective object code modules, the routine having an associated name and each implementation adapted to a selected hardware configuration;

means for associating the object code modules with the name of the routine and respective sets of hardware characteristics; and

means for resolving when the application program is loaded into memory of the computer system, a reference to the routine using the sets of hardware characteristics and a hardware configuration of the system.

14. (original) A computer-implemented symbol table for referencing a library of object code modules that implement a plurality of routines, comprising:

a first set of one or more entries, each entry in the first set including a unique name of a routine and a reference to an object code module in the library; and

a second set of one or more entries, each entry in the second set including a shared name of a routine, a set of hardware characteristics, and a reference to an object code module in the library.

15. (original) The symbol table of claim 14, wherein the hardware characteristics include at least one of clock speed of a processor, processor model, cache configuration, hardware operation latency times, instruction set characteristics, bypass characteristics, branch prediction behavior, pre-fetching capability, information describing stall conditions, branch penalties, size and associativity of processor data structures, queue sizes for out-of-order or decoupled processors, and the number of processors in a multi-processor system.

16. (original) A computer program product configured for causing a computer to perform the steps of:

compiling a plurality of implementations of a routine into respective object code modules, the routine having an associated name and each implementation adapted to a selected hardware configuration;

associating the object code modules with the name of the routine and respective sets of hardware characteristics; and

resolving when the application program is loaded into memory of the computer system, a reference to the routine using the sets of hardware characteristics and a hardware configuration of the system.